JOURNAL OF INFORMATION TECHNOLOGY AND ITS UTILIZATION, VOLUME 3, ISSUE 1, JUNY-2020, 9-13
ISSN 2564-802X

RABIN-CARP IMPLEMENTATION IN MEASURING SIMALIRITY OF RESEARCH PROPOSAL OF STUDENTS

Herman[1], Lukman Syafie[2], Tasmil[3], Muhammad Resha[4]

[12]Computer Science Dept. Universitas Muslim Indonesiaut, Indonesia
[3]Balai Besar Penelitian dan Pengembangan SDM KOMINFO Makassar, Indonesia
[4]STMIK AKBA Makassar, Indonesia
[1]herman@umi.ac.id,[2]lukman.syafie@umi.ac.id,[3]tasmil@kominfo.go.id,[4]m.resha@gmail.com

***Abstract*-- Plagiarism is the use of data, language and writing without including the original author or source. The place where palgiate practice occurs most often is the academic environment. In the academic world, the most frequently plagiarized thing is scientific work, for example thesis. To minimize the practice of plagiarism, it is not enough to just remind students. Therefore we need a system or application that can help in measuring the level of similarity of student thesis proposals in order to minimize plagiarism practice. In computer science, the Rabin-Karp algorithm can be used in measuring the level of similarity of texts. The Rabin-Karp algorithm is a string matching algorithm that uses a hash function as a comparison between the search string (m) and substrings in text (n). The Rabin-Karp algorithm is a string search algorithm that can work for large data sizes. The test results show that the use of values on k-gram has an effect on the results of the measurement of similarity levels. In addition, it was also found that the use of the value 5 on k-gram was faster in executing than the values 4 and 6.**

***Keywords*: plagiarism; thesis proposal; rabin-karp.**

I. INTRODUCTION

The rapid development of information technology makes its users easily obtain data and information. Data and information can be in the form of text, images, audio or video. However, this has a negative impact, which is easily plagiarism. According to [1], the problem of plagiarism is increasing because of the digital era.

Plagiarism is the use of data, language, and writing without including the original author or source [1]. According to the Regulation of the Minister of National Education of the Republic of Indonesia No. 7 of 2010, plagiarism is the act of intentionally or unintentionally in obtaining or trying to obtain credit or value by quoting part or all of the scientific work of others without stating the exact and adequate source. According to [2], the academic world is the place where plagiarism is most common.

Indonesia, a developing country, is not the only one affected by palgiarism. According to [3], several cases of palgiat were also found in developed countries. The place where palgiate practice occurs most often is the academic environment. In the academic world, the most frequently plagiarized thing is scientific work, for example thesis. To minimize the practice of plagiarism, it is not enough to just remind students. Therefore we need a system or application that can help in measuring the level of student thesis similarity in order to minimize the practice of plagiarism[3].

In computer science, the Rabin-Karp algorithm can be used in measuring the level of similarity of texts. The Rabin-Karp algorithm is a string matching algorithm that uses a hash function as a comparison between the search string (m) and substrings in text (n). If the values of the two hashes are the same then a further comparison will be made of the characters. Comparisons are made from left to right (n-m) times [4]. According to [5], that the Rabin-Karp Algorithm is a string search algorithm that can work for large data sizes. It is on this basis that the Rabin-Karp Algorithm is used in this research.

The use of the Rabin-Karp Algorithm has been carried out by [6]. In that study the Rabin-Karp Algorithm is used to optimize heterogeneous solutions that take into account the benefits and limits of achieving Online Analytical Processing (OLAP) in terms of response time. The results of the study showed better performance in terms of response time and memory usage.

Some researchers have conducted research related to string matching, including [1] using architecture and algorithms to detect case-copy or plagiarism. Dayarathne and Ragel [5] conducted research using the Rabin-Karp Algorithm in finding the appearance of a pattern in the text of a collection of strings. The same thing was done by [7] and [8], using the Rabin-Karp Algorithm in matching text or string patterns. Rasywir et.al [9] use the Rabin-Karp Algorithm to automatically evaluate students' essay answers. On this basis, the researcher used the Rabin-Karp algorithm to measure the level of similarity of student research proposals.

II. METHODE

The proposed system design of this study consists of 2 parts, namely data collection and measurement of the level of similarity. The first step taken is to collect data on student proposals that already exist in the database and then proceed to measure the level of similarity of student research proposals that are entered and compare with student research proposal data that has been stored in a database. Proposal data in the form of text that will be stored in a database or similarity level must be measured through preprocessing, where this process

consists of letter folding, tokenizing, filtering, and stemming. The measurement results of the similarity level are then displayed for each proposal data that has been stored in the database. An overview of the proposed system is shown in Figure 1.

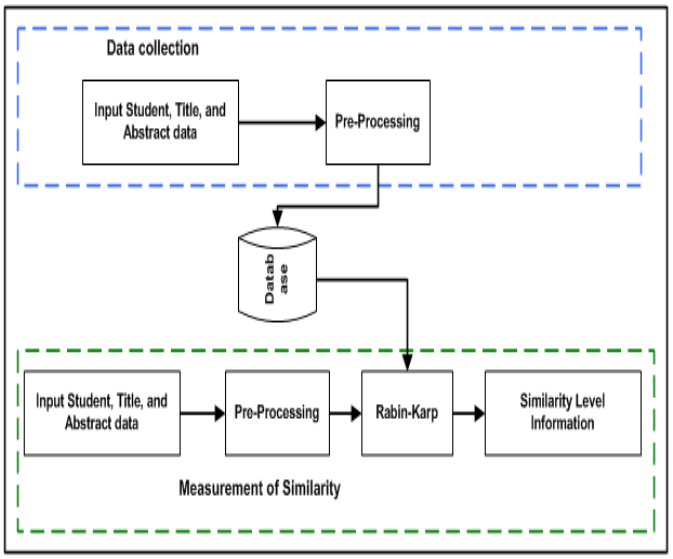

Fig. 1. System design

The data that will be processed in this study is student research proposal data in the form of abstract. The data is inputted and stored in a database, and then encoded in the form of numbers to facilitate the process of calling data. The results of the Rabin-Karp process are then displayed in the form of a list that displays the similarity values of each proposal data stored in the database. An overview of data processing can be seen in Figure 2.

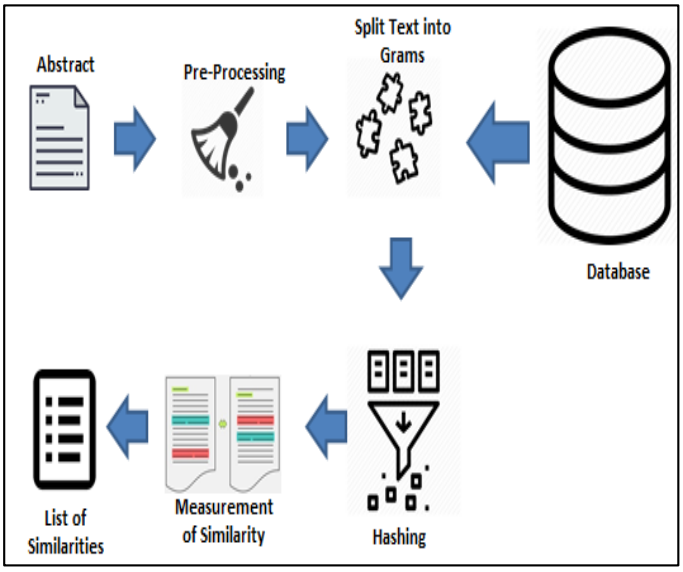

Fig. 2. Data Processing using Rabin-Karp

An illustration of the implementation of the Rabin-Karp algorithm as a detailed description of Figure 2 is as follows:

1. Remove the punctuation marks and change it to source text and words we want to find into words without capital letters. Example in Figure 3.

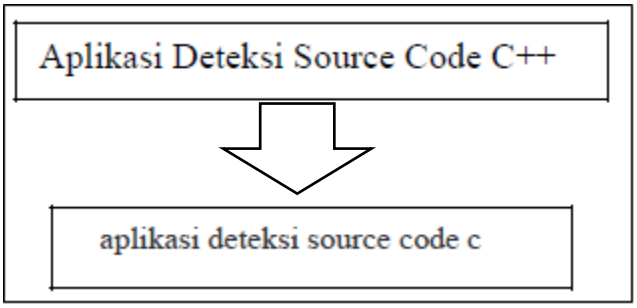

Fig. 3. Example text (Source : [4])

2. Divide the text into grams that are determined by the k-gram value. For example the size of k = 7 then the results are like in Figure 4.

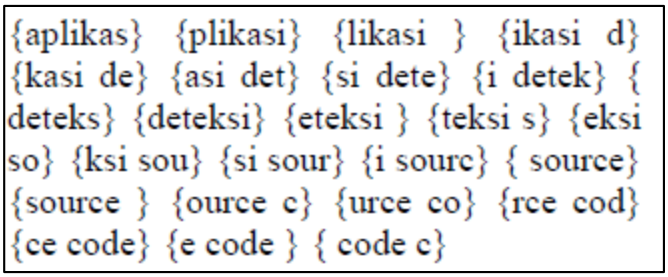

Fig. 4. Gram yields with k = 7 (Source : [4])

3. Look for a hash value with the rolling hash function of each gram formed as in Figure 5.

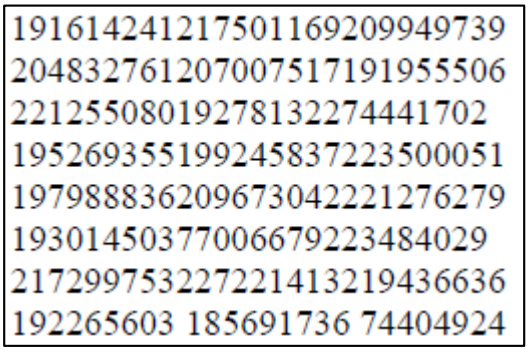

Fig. 5. Value of each gram (Source : [4])

4. Look for the same hash value between the two texts as in Figure 6.

191614241	217501169	209949739
204832761	207007517	191955506
221255080	192781322	74441702
195269355	199245837	223500051
197988836	209673042	221276279
193014503	77006679	223484029
217299753	227221413	219436636
192265603	185691736	74404924

Fig. 6. A hash value of two texts (Source : [4])

5. Determine the similarity value of two texts with the Dice's Similarity Coefficient equation.

To calculate similarity, Dice Similarity Coeficients can be used. By counting the number of K-Grams used in the two texts tested. The similarity value can be calculated with equation (1).

$$S = \frac{2C}{A+B} \quad (1)$$

Where :
S = Value of similarity
C = same amount of K-Gram
A and B = Number of K-Grams of each string

III. Result and Discussion

In this study, testing was carried out with two approaches. The first approach is the exploration of k-gram values. The k-gram values tested were 4, 5, and 6. These values are based on several previous studies, such as those conducted by Hartanto, et al (2019). Even though the k-gram value is the same, the test method is different. In a study conducted by Hartanto, et al (2019), tested 4 thesis abstracts that have not been stored in the database. In this research, all thesis abstracts that have been stored in a database are tested to see whether the Rabin-Karp algorithm is appropriate in measuring the similarity of students' abstract theses. In addition, also to see the effect given if the k-gram value changes. The second approach is to test the time needed to measure the resemblance of students' abstracts from each k-gram value. In Table 1 shows the similarity measurement results using the value 4 in k-grams and Figure 7 shows the graph of the measurement results.

In Table 1 shows each abstract whose measured similarity produces a value of 100 in the same abstract. For example, document number 1 compared to document number 1, produces a value of 100. Document number 1, compared to document number 2, produces a value of 8.33. A value of 100 is a similarity percentage value of the document being compared, as is the value of 8.33 and other values. However, documents 4 and 9 yield a similarity value of 100 which indicates that the two documents are the same. Actually documents 4 and 9 have different abstracts. This indicates that the measurement of similarity with a value of 4 on k-gram is still inaccurate.

TABLE I
SIMILARITY MEASUREMENT RESULTS USING THE VALUE 4 IN K-GRAM

Document Number	01	02	03	04	05	06	07	08	09	10
01	100	8.33	18.18	59.26	20.51	17.14	22.86	21.43	59.26	20.29
02	8.33	100	9.76	15.38	16	9.52	7.14	19.05	15.38	10.91
03	18.18	14.63	100	9.09	17.86	30.77	25.29	30.14	9.09	48.84
04	59.26	15.38	9.09	100	21.43	8.33	16.95	13.33	100	13.79
05	20.51	16	17.86	21.43	100	11.11	19.72	14.04	21.43	28.57
06	22.86	9.52	26.92	8.33	11.11	100	11.94	26.42	8.33	27.27
07	34.29	10.71	34.48	23.73	30.99	8.96	100	18.18	23.73	55.45
08	28.57	19.05	32.88	13.33	14.04	33.96	20.45	100	13.33	36.78
09	59.26	15.38	9.09	100	21.43	8.33	16.95	13.33	100	13.79
10	23.19	10.91	44.19	13.79	34.29	30.3	43.56	36.78	13.79	100

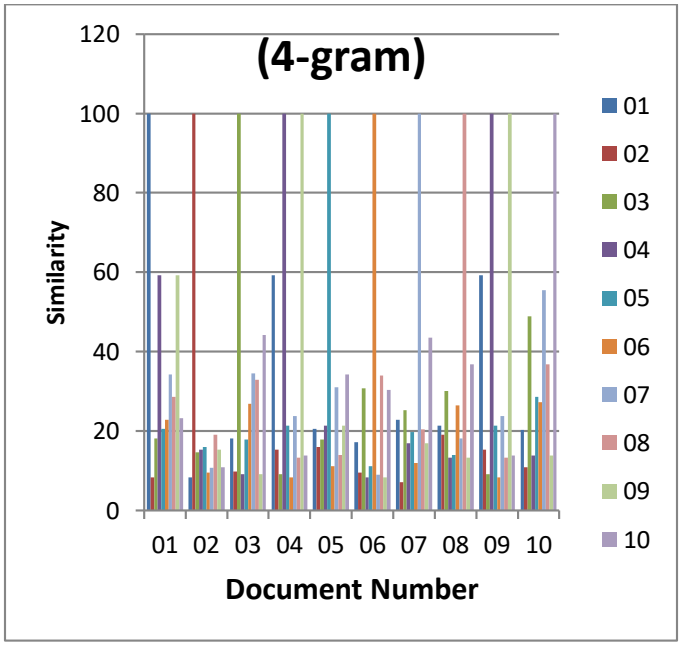

Fig. 7. Chart similarity measurement results using the value 4 in k-grams

Then the test is done using the value of 5 on k-gram. In Table 2 and Figure 8 displays the results of similarity measurements for each student abstract thesis document.

Measurement values for each abstract document compared to the same document still produce a value of 100. In documents 4 and 9 they no longer produce the same value or are no longer considered the same document. However, in this test, some measurement results display values above 100, whereas measurement values should not exceed 100.

TABLE II
SIMILARITY MEASUREMENT RESULTS USING THE VALUE 5 IN K-GRAM

Document Number	01	02	03	04	05	06	07	08	09	10
01	100	52.25	51.61	78.21	44.44	75.41	68	63.09	69.52	64.15
02	116.04	100	47.6	114.39	58	102.22	63.86	106.56	76.79	85.96
03	30.11	15.1	100	31.96	19.35	46.03	21.95	19.54	36.96	29.79
04	85.26	55.42	57.73	100	38.3	85.71	43.27	67.09	75.23	63.64
05	26.67	12.53	22.58	23.4	100	25	36.84	15.2	18.6	27.27
06	61.48	36.77	55.56	57.94	35	100	44.29	45.32	50.67	56.58
07	44	19.96	29.27	32.69	31.58	40	100	24.31	32.08	44.44
08	104.29	85.08	59.2	110.55	43.86	100.49	57.46	100	74.73	86.63
09	45.71	24.73	41.3	47.71	18.6	45.33	32.08	31.18	100	52.54
10	42.45	25.92	40.43	46.36	31.82	50	37.04	31.02	52.54	100

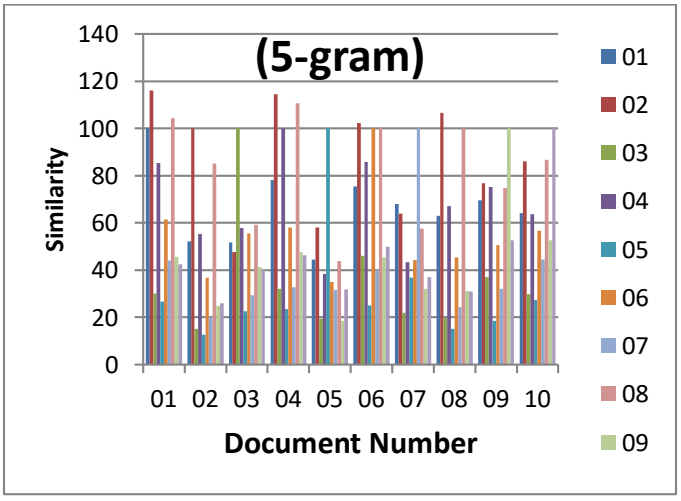

Fig. 8. Chart similarity measurement results using the value 5 in k-grams

Next is to test with a value of 6 on k-gram. Table 3 and Figure 9 show the results of measuring the similarity of student proposal abstracts. The measurement results for each abstract document are still the same as Table 1 and Table 2. The results of this test no longer display values above 100 and documents 4 and 9 have been considered as different documents. If we look at Figure 9, some values of measurement results exceed 50, some even above 60. But this does not indicate the results of measurement of incorrect similarity. The results of testing the first approach show changes in the value of the k-gram gives an effect on the similarity measurement results.

TABLE II
SIMILARITY MEASUREMENT RESULTS USING THE VALUE 6 IN K-GRAM

Document Number	01	02	03	04	05	06	07	08	09	10
01	100	29.17	12.12	36.14	25	26.32	22.95	25.17	34.29	31.46
02	41.67	100	14.46	45.11	50.68	54.88	54.05	51.81	43.87	41.73
03	12.12	14.46	100	20	14.46	15.84	16.67	13.85	10.87	7.89
04	55.42	49.62	22.86	100	43.61	47.68	42.86	41.11	66.2	53.97
05	31.25	47.95	16.87	42.11	100	46.34	39.64	60.1	46.45	48.92
06	33.33	59.76	25.74	47.68	57.32	100	34.11	60.66	64.74	53.5
07	36.07	45.05	12.5	36.73	36.04	29.46	100	36.71	41.67	34.62
08	36.36	69.43	18.46	51.11	69.43	71.09	51.9	100	67.33	63.44
09	57.14	50.32	8.7	73.24	60.65	62.43	43.33	57.43	100	63.51
10	38.2	44.6	7.89	49.21	51.8	50.96	38.46	50.54	56.76	100

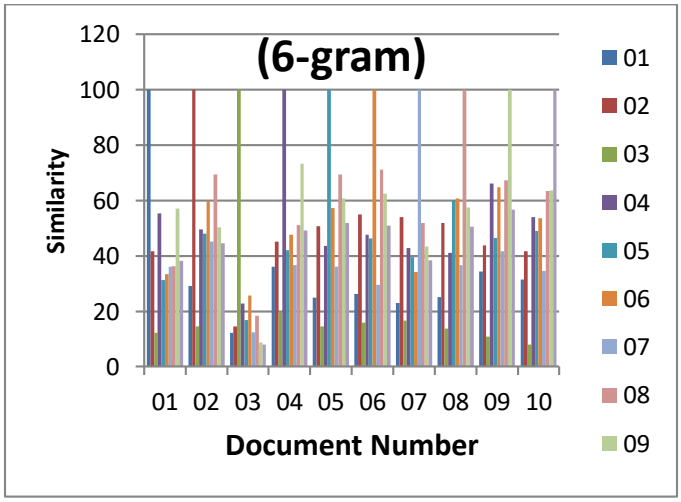

Fig. 9. Chart similarity measurement results using the value 6 in k-grams

Next is testing the second approach. This test aims to see the time needed to measure the similarity for each value on k-gram. Figure 10 shows the time taken and it appears that the value of 5 in k-gram takes less time than the values of 4 and 6 in k-gram. When compared again with the similarity

measurement results, the value of 5 on k-gram indicates an inaccurate measurement level.

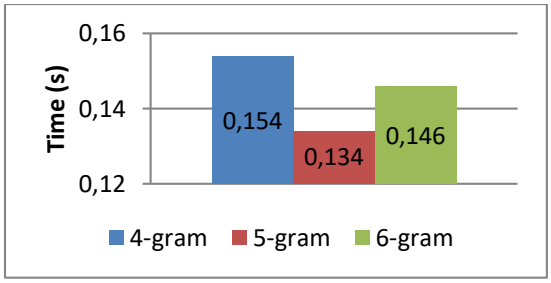

Fig. 10. Testing with the execution time approach

IV. Conclusion

From the results of this study, several conclusions were obtained, namely:

1. The k-gram value has an effect on the measurement results of the abstract similarity of the proposal.
2. A value of 5 on the k-gram requires less time to measure the similarity of the abstract proposal.

V. Acknowledgment

This research cannot be completed without help from other parties. therefore, the authors thank the Faculty of Computer Science for morally supporting and infrastructure. Acknowledgments also the author gave to the Universitas Muslim Indonesia for providing financial assistance so that this research can be implemented.

VI. References

[1] J. Agarwal *et al.*, "Intelligent plagiarism detection mechanism using semantic technology: A different approach," *Proc. 2013 Int. Conf. Adv. Comput. Commun. Informatics, ICACCI 2013*, pp. 779–783, 2013.

[2] G. Acampora and G. Cosma, "A Fuzzy-based approach to programming language independent source-code plagiarism detection," *IEEE Int. Conf. Fuzzy Syst.*, vol. 2015–Novem, 2015.

[3] A. Yudhana, A. D. Djayati, and Sunardi, "Sistem Deteksi Plagiarisme Dokumen Karya Ilmiah dengan Algoritma Pencocokan Pola," *Jurti*, vol. 1, no. 2. pp. 178–187, 2017.

[4] F. T. Informasi *et al.*, "PERBANDINGAN ALGORITMA WINNOWING DENGAN ALGORITMA RABIN KARP UNTUK MENDETEKSI," vol. 8, no. 3, pp. 124–134, 2017.

[5] N. Dayarathne and R. Ragel, "Accelerating Rabin Karp on a Graphics Processing Unit (GPU) using Compute Unified Device Architecture (CUDA)," *2014 7th Int. Conf. Inf. Autom. Sustain. "Sharpening Futur. with Sustain. Technol. ICIAfS 2014*, 2014.

[6] H. I. M. Alzeini, S. A. Hameed, and M. H. Habaebi, "Optimizing OLAP heterogeneous computing based on Rabin-Karp Algorithm," *2013 IEEE Int. Conf. Smart Instrumentation, Meas. Appl. ICSIMA 2013*, no. November, pp. 26–27, 2013.

[7] O. S. Joshi, B. R. Upadhvay, and M. Supriya, "Parallelized Advanced Rabin-Karp Algorithm for String Matching," *2017 Int. Conf. Comput. Commun. Control Autom. ICCUBEA 2017*, pp. 1–5, 2018.

[8] L. S. N. Nunes, J. L. Bordim, Y. Ito, and K. Nakano, "A Prefix-Sum-Based Rabin-Karp Implementation for Multiple Pattern Matching on GPGPU," *Proc. - 2018 6th Int. Symp. Comput. Networking, CANDAR 2018*, pp. 139–145, 2018.

[9] E. Rasywir, Y. Pratama, Hendrawan, and M. Istoningtyas, "Removal of modulo as hashing modification process in essay scoring system using rabin-karp," *Proc. 2018 Int. Conf. Electr. Eng. Comput. Sci. ICECOS 2018*, vol. 2019–Janua, pp. 159–164, 2019.